App Project 2- Bring Your Own Idea (BYOI)

Assigned: 10/28
Due Dates: See Schedule section
Project Type: Individual or 2-Person Team
Submission: See Deliverables section

Description

For this project, you will be asked to submit your own idea which fulfills a number of loose requirements. I will approve your idea and also help you scope it such that it's a reasonable project for a 4-5 week period. On our final lecture, you will demo your project to the class.

Schedule

The remaining schedule of topics this semester, with the Project 2 dates.

Week	Date	Topic / Deliverable
10	10/28	Permissions, Location, Handling Rotation **Project 1 due (11/03), Project 2 assigned**
11	11/04	Firebase: Authentication, Realtime DB **Project 2 proposals due (11/08)**
12	11/11	Firebase: Realtime DB, Analytics, Crash Reporting **Quiz 3**
13	11/18	Image Loading, Notifications **Project 2 check-in #1**
14	11/25	Build Variants, Code Obfuscation, Decompilation
15	12/02	Sensors **Quiz 4, Project 2 check-in #2**
16	12/09	Final Project Presentations & Demo **Project 2 code due (12/13)**

What You'll Need

- The only lecture required to do this project is Lecture 11 (Firebase Auth).
- **Claim your GitHub repo**: https://classroom.github.com/g/ZgCZp6My
    - Create (or join) a team, even if solo. Team name = the name of your app.

- Your private repo will be created with a readme, so you can start by cloning it.

Requirements

Idea Requirements

Your idea does NOT have to be original! You may pick something that exists in some capacity already (like an app that would've tracked the World Series scores, even though the MLB app exists).

- Your app must have at least **four Activities:** a **LoginActivity** and **SignUpActivity** (some of which we will build in class), **plus two** other Activities.
  - You will use Firebase Authentication to implement login / sign up.
  - You may combine the Login & Sign Up screens together by properly showing / hiding views to switch UI experiences, if desired.
  - If you have a project idea that requires user authentication or OAuth with another service (e.g. Twitter user login), then we can talk about adjusting or accommodating this requirement.
- Your app must use **at least one API** endpoint that we haven't used in class / Project 1.
  - e.g. you may use Twitter's other APIs, but the Search Tweets API will not count.
- Your app must use **one** other "complexity".
  - These are listed on the next page.
- Your app must support at least **two** languages (English plus one other language).
  - You do **not** need to translate dynamic information (e.g. if you're displaying content returned from an API). Just built-in / hard-coded strings need to be translated (e.g. button text / app bar titles / etc.)

If you are doing something *"reasonably" complex* that's not covered, you can run it by me and I may count it towards either the API call requirement or the complexity requirement.

API Ideas

- https://github.com/toddmotto/public-apis
- https://any-api.com/

Make sure you confirm that the API you plan to you is free for a reasonable amount of use. As opposed to being free for a period of time or until you hit a certain number of calls.

Complexities Options

- **Another API call**
    - OAuth does not count.
    - Can be the same source / company as the 1st API call, just a separate endpoint.
    - Some companies offer SDKs to interface with their APIs, this is also acceptable.

- Any Android component that requires a **permission prompt** (**except** location, unless different from how we used it in lecture).

- **Firebase Realtime DB**, or Firebase products of a *similar complexity* (e.g. Cloud Storage).

- Usage of another **Google Play Services SDK** we haven't seen in class (e.g. Google Drive, Google Places, Google Fit, etc.)
    - Google Maps *could be* okay if different from lecture / Project 1.

- Some projects in the past have also used features like Bluetooth or hardware sensors - e.g. something new we haven't learned in class or more advanced usage of something we have learned in class.

Project Requirements

- **Min SDK:** at least API 21
- **Use String resources wherever possible!!** If you need to load them at runtime (e.g. for a Dialog or a Toast), use the getString(R.string.your_string) function.

Git Requirement

Part of your grade will be your utilization of Git / GitHub. Prefer small, incremental commits. Work in branches and create PRs for major pieces of work (for example, one per screen). Include a helpful description in the PR and merge it yourself. I may come in during grading to leave comments & feedback. **Do not check in your private API keys into GitHub!** Use one of the techniques we will discuss in class -- .gitignore, local.properties, etc.

\

Detailed Requirements

Requirement	Description
Login	Your login screen show allow the user to enter in a email / password. Use Firebase Authentication to log the user in. Include a button for the user to go to a Sign Up Activity to register for a new account **or** show / hide views to switch to a Sign Up UI if you want to stay in the same Activity.
Remember Credentials	Include two switches to remember user credentials (email and password) and pre-fill the EditTexts if remembered. If the email and/or password are being remembered, the switches should be turned on by default. **Note** - The FirebaseAuth SDK will technically "keep the user logged in", but I'd like you guys either prompt the user for credentials again or automatically skip to the next screen in this scenario - up to you.
Sign Up	Allow the user to give an email and password to sign up. You do not have to send a verification email. **You should have the user enter in their password twice** (e.g. a Password field and a Confirm Password field). *Do not allow them to sign up if the two passwords don't match.* Use Firebase Authentication to register and store user credentials.
Localization	Include support for both English and one other language of your choice. Translations do not have to be accurate -- feel free to use Google Translate. You do **not** need to translate dynamic information (e.g. strings returned by an API), but the strings that you include in your strings.xml file should all have translated counterparts.

Deliverables

Proposal Submission

Due **11/08 by 11:59 PM**. Submit a text document which includes ***at least one idea***:

- Whether you will work solo or who your partner will be (only one submission needed for a group).
- A paragraph (e.g. 4-5 sentences) describing your project idea.
- A description of what your extra screens will be.
- Links to the specific API(s) and complexities you plan to use.

IMPORTANT: If applicable, create a developer account for the APIs you plan to use and **confirm** that you receive an API Key / Secret, make sure the APIs are "free" (e.g. not a trial) or a reasonable amount of use. **You are responsible for choosing reliable APIs for your project.**

Check-in #1

Due **11/18 by 11:59 PM**, we will have our first check-in. You **must** have completed:

- Either your API call or your "complexity" choice.
    - Includes the accompanying screen.
    - I'll accept a *significant* amount of progress in lieu of it being completely done (e.g. most of it is done, but maybe there's a bug you're stuck on).

For the check-in, just make sure you have submitted at least one PR & merged it into your private repo. **Submit the PR link on Blackboard!**

Check-in #2

Due **12/02 by 11:59 PM**, we will have our second check-in. You **must** have completed:

- Either your API call or your "complexity" choice (whichever wasn't done for the 1st check-in).
    - Includes the accompanying screen.
    - I'll accept a significant amount of progress in lieu of it being completely done (e.g. most of it is done, but maybe there's a bug you're stuck on).

For the check-in, just make sure you have submitted at least one PR & merged it into your private repo. **Submit the PR link on Blackboard!**

Final Presentation / Demo

Due **12/09; can not** be submitted late. On our final lecture day, we will conduct presentations & demos of each project. Each team will get **5 minutes** to present, **plus 1 minutes** of Q&A from myself or other students. You should explain the API / complexities used and the major screens in your app. Additionally, share at least one interesting technical detail (maybe you used a new library or Android functionality; maybe an API was interesting; etc.)

You are also **required to submit a video recording** of your app working (there is a screen record feature built into Android Studio), in case there is an issue with your project or presentation.

Final Submission

Due on 12/13 by 11:59 PM.

On **GitHub**:

- **Make sure your code is pushed up / merged into the master branch by the deadline.** Make sure you have been commit code often, using PRs, and no API keys committed!

On **Blackboard** submit:

- Your **debug** APK file.
    - Use the Gradle tab on the right side of Android Studio and choose *Build -> assemble* under your app module.
    - After completion, the APK will be found under *<your app module>/build/outputs/apk/debug/<apk name>.apk*
- A short document which contains:
    - Special instructions / bugs / issues / limitations you are aware of.
- A **video** of your application working (in its ideal scenario)
    - I'd prefer a Google Drive link, for space (make sure it's properly shared).
- A **screenshot** of **each screen** in your application.
- **Screenshot(s)** of your Firebase dashboard, showing:
    - Authentication (e.g. user sign-ups), Database (if used for a complexity)

Grade Breakdown

Coming soon!